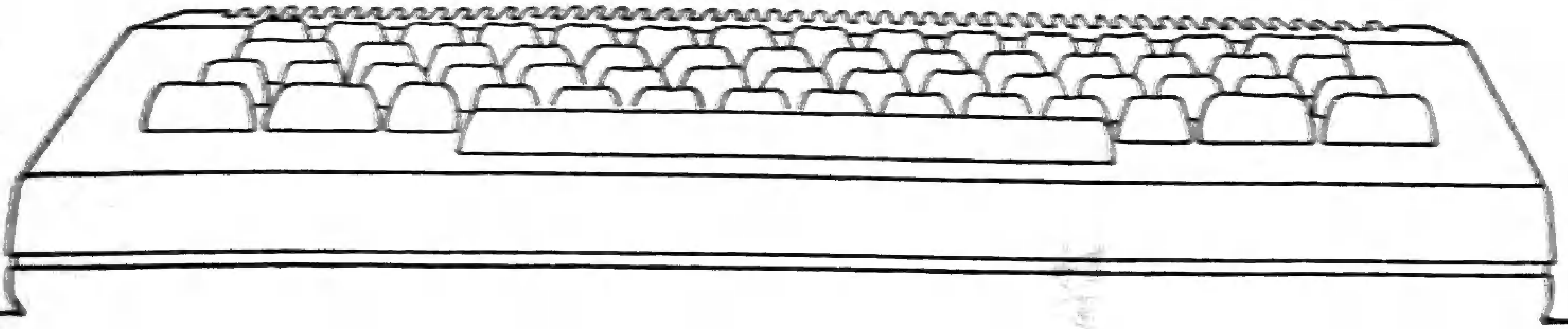
CAMPUTERS

LYNX

LYNX DOS



CAMPUTERS
LYNX

DOS MANUAL

CONTENTS Page

1 **Introduction**

How to use this guide....................................1
The essentials...1
Hardware checklist.......................................2

2 **What is a DOS?**

Disk and cassette systems 3

3 **Getting Started**

Assembling the disk drive and interface box.............4
Powering up procedure....................................5
Initialisation...5
Some basic DOS commands..................................6
Powering down..9

4 **The DOS Commands**

The DOS commands and minimum abbreviations..............10
Command descriptions and examples.......................11

5 **Disk Care**

Looking after floppy disks..............................21
Write protection..22
Recommended disks.......................................22

6 **Technical Appendix**

Initialisation routine..................................23
The disk directory......................................24
Disk structure..24
The Lynx DOS header.....................................24
CP/M - an alternative DOS...............................25

7 **Error Messages and Error Codes**

Syntax errors...26
Error messages..26
Error codes...27

8 **Lynx DOS Reference Card**

Chapter 1: INTRODUCTION

HOW TO USE THIS GUIDE

This is a preliminary release of the Lynx Disk Operating System Manual.

Before you start using the disk drives attached to the Lynx, please follow the instructions in this introduction - even if you are very familiar with disk systems in general. If you switch on the mains power to the components in the wrong order, for instance, the system will not be initialised correctly and the DOS commands will not work.

We appreciate the fact that you may have just unpacked the disk drive and disk drive interface box and be very keen to get started. You may feel that you have no need to read the manual all the way through - why not just try a few things out and see how it goes?

We agree: this is a good way to learn. You do not have to read this manual from beginning to end, but you must, at the very minimum, read the following essentials.

THE ESSENTIALS

		see pages
1.	Check that you have all the hardware items necessary.	2
2.	If you have not used diskettes before, then read Chapter 5. If you mishandle them (by touching them in the wrong place, for instance), you can make then completely unusable.	21 to 22
3.	Connect the hardware together properly and switch the units on in the correct order. If you don't, none of the Lynx DOS commands will work.	4 to 5
4.	There is a brief 'beginner's session' given in Chapter 3. After you have completed this, you can explore the system on your own - at your own pace.	5 to 9

THAT'S ALL!

The rest of this guide forms a reference manual which can answer many of your queries as they arise.

After you have completed the above "checklist", the next most essential part of the manual is probably Chapter 4. This describes all the extra DOS commands in full and in alphabetical order.

Some errors will occur in everyday use (by mis-typing a command, for instance) and others can only be provoked with great ingenuity. You will probably only want to refer to Chapter 7 (Error messages and error codes) as the errors are reported.

The Technical Appendix contains various related items of information which are assembled in Chapter 6 to avoid cluttering the text for the casual user.

The final page of the manual is a summary of the Lynx DOS commands and their syntax, in reference card format. By "syntax" we mean the order in which the parts of a command have to be given in order to be interpreted correctly. You may photocopy this page, or cut out the original, to keep near the Lynx as a handy reference. Even experienced users sometimes forget the names of commands and the extra words of letters (called "arguments") which are needed to complete them.

HARDWARE CHECKLIST

The following items of equipment are needed to complete a Lynx disk system:

- This manual!
- A 96K or 128K Lynx microcomputer (or a 48K Lynx upgraded to a 96K model)
- A 40- or 80-track Lynx disk drive unit with mains lead and plug
- A Lynx disk drive interace box
- A length of 26-way ribbon cable with connectors at each end
- At least 5.25" one single sided, double density, soft sectored 40-track floppy disk (for the 200K drive). Double sided, quad density disks are required for the 800K drive.

All these items should be available from your Lynx dealer.

Keep the original packing, in case you need to return any hardware. If there is a fault, CONTACT YOUR DEALER FIRST. Do not send the disk system to Camputers until you have checked with us, there may be a simple remedy.

Chapter 2: WHAT IS A DOS?

"DOS" is an abbreviation of "Disc Operating System". Lynx DOS is a machine code program which, at the lowest level, controls the exact position and the action of the read/write head over the disk. At a more familiar level, it provides extra commands available from the keyboard. These commands carry out tasks which involve reading and writing to files, whilst automatically looking after individual disk files and complete disks.

DISK AND CASSETTE SYSTEMS

As a Lynx owner, you have probably used the Lynx to load programs - perhaps games programs - from cassette tape. You may have progressed to storing BASIC or machine code programs that you have entered into the Lynx's memory (RAM) from the keyboard. In this case, you will be familiar with the built-in Cassette Operating System, which allows you to store a BASIC program on tape from RAM (an operation called SAVEing), or have LOADed a program on tape into the Lynx's memory. Related tape commands, such as APPEND, MLOAD, TAPE and VERIFY make up the rest of the Cassette Operating System.

When you use a disk system, you will need to do these same or similar operations. In all, nineteen commands have been added which allow you to create, delete, rename, add comments to, protect, append and display information - on individual files or collections of them. Some of these new commands are introduced in the next Chapter, and are more fully described in Chapter 4 ("The DOS Commands"). Commands like SAVEing and LOADing to and from disks are similar to the cassette counterparts, except that the minutes that a cassette recorder can take are reduced to seconds with disks.

A further bonus is that because disk units are precision engineered, files are transferred with much greater reliability. The storage capacity of an individual disk ranges from 200 Kbytes (for a single sided 40-track disk drive) to 800 Kbytes (for a double sided 80-track drive) - and because disks are completely flat, you can store lots of them in a very small storage space. The maximum number of files which can be saved on a single disk is just under 800 - and you can access any one of these files in under one second!

Once you have become familiar with a disk system, you will be most reluctant to return to using a tape storage medium again!

Chapter 3: GETTING STARTED

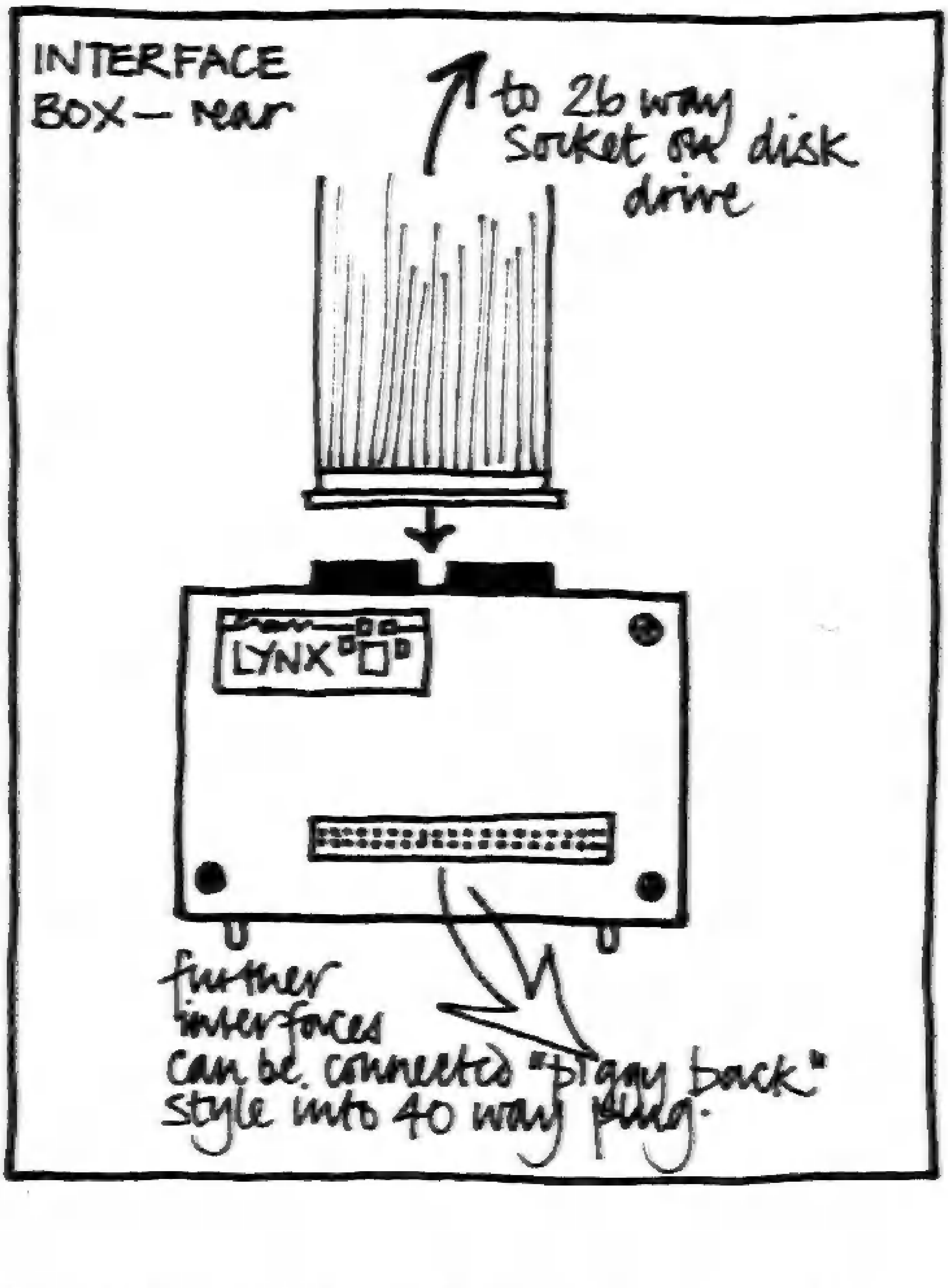
ASSEMBLING THE DISK DRIVE SYSTEM

In addition to a 96K or 128K Lynx microcomputer, you will need three further components: the disk drive itself, an interface box (which contains the Disk Operating System ROM) and a cable with connectors at each end. Plugging the components together is an easy task, since there are only three connections to make.

Follow the normal procedure for connecting computer modules together and DISCONNECT ALL SOURCES OF ELECTRICAL POWER BEFORE YOU PLUG OR UNPLUG ANY EXTERNAL COMPONENT.

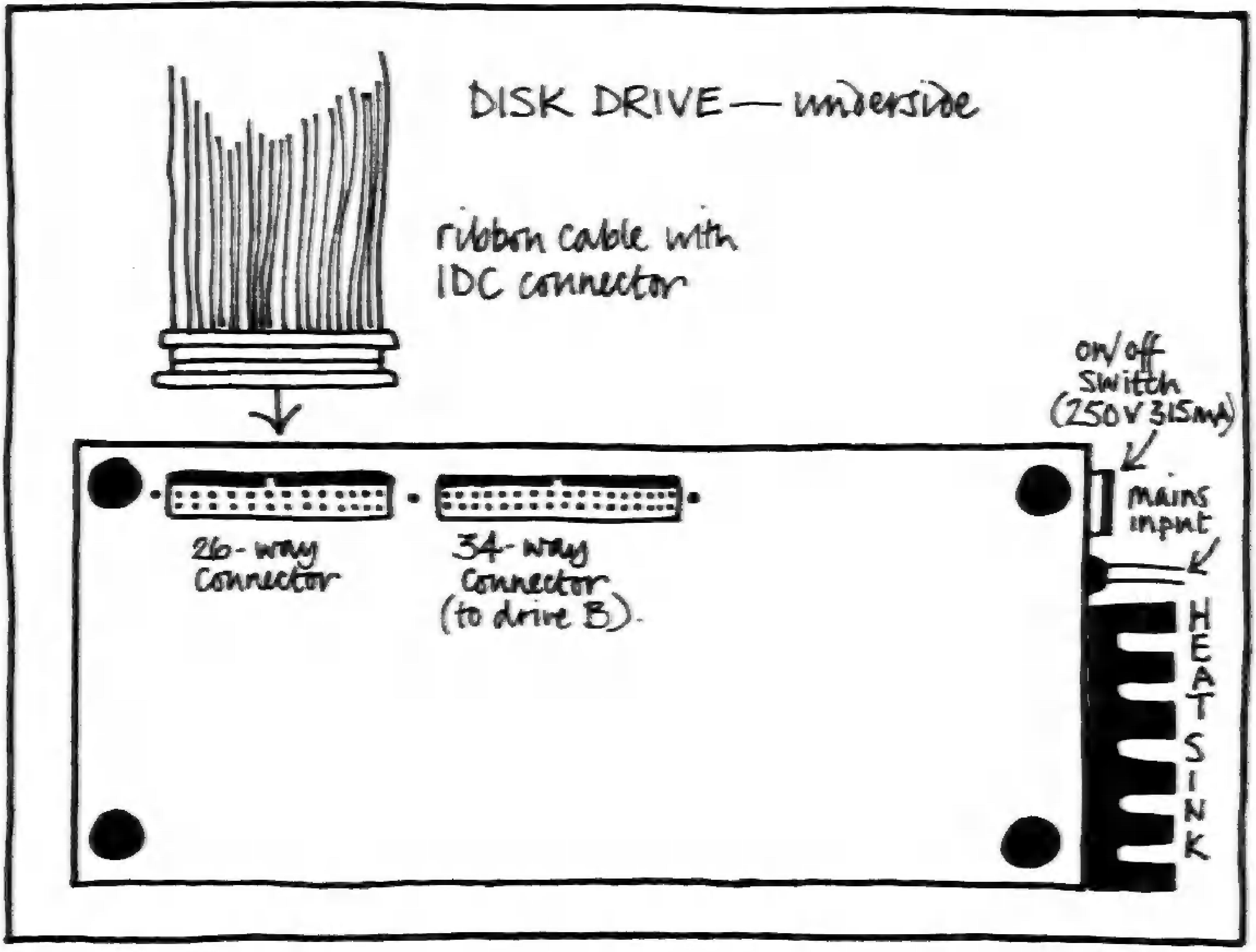
You may also be supplied with a 5.25" floppy disk, if not, ask your dealer to reccommend the correct type of disk for the unit. Read Chapter 5 for details - especially if you are not familiar with disks, since can you make them permanently unusable if they are mishandled.

The steps to follow in connecting the components are:


INTERFACE BOX — rear
to 26 way Socket on disk drive
LYNX
further interfaces can be connected "piggy back" style into 40 way plug.

1. Plug one end of the ribbon cable connector into the top of the interface box. It doesn't matter which end of the ribbon cable you choose - and the 26-way connector is 'polarised', so you can't, except by using extreme force, plug it in the wrong way round.

2. Connect the other end of the ribbon cable to the underside of the disk drive (as in the diagram, overleaf)



3. Carefully position the interface box 40-way socket over the INTERFACE plug at the rear of the Lynx. Gently ease the box by a slight rocking motion until the box is fully pushed in to the Lynx.

The disk system is now connected and is ready for switching on - but this must be done in the correct order.

POWERING UP PROCEDURE

4. The disk unit should be plugged into the mains and switched ON (an illuminated rocker switch is at the rear of the unit, to the right of the black heat sink). A red LED (Light Emitting Diode) should come on at the front of the unit, to show that the drive has been 'selected'.

5. Plug in and switch on the Lynx in the usual way. You should hear the familiar two 'beeps', followed by the Lynx logo and prompt.

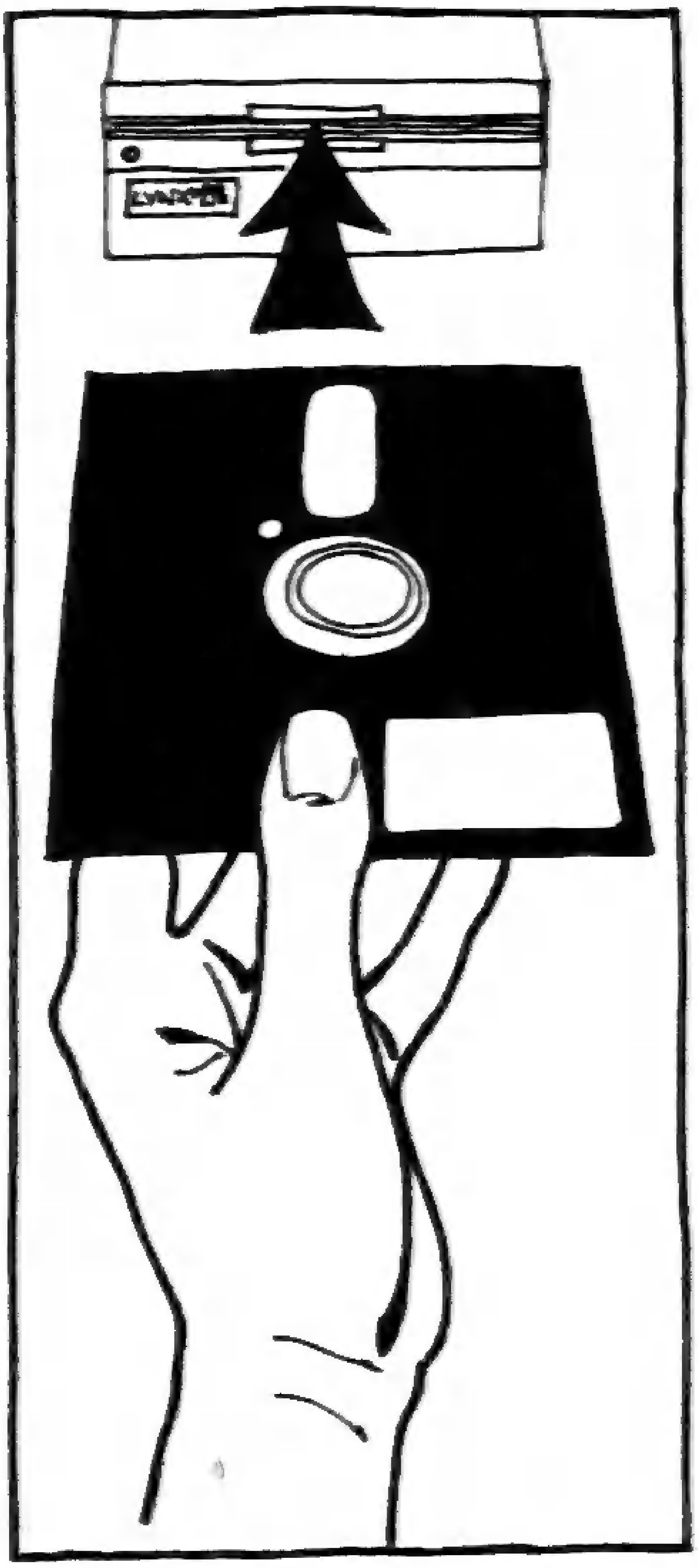
INITIALISATION

You are now ready to load the Disk Operating System software into the Lynx's memory. This is easily done by using the external ROM command.

6. Type:

```
        XROM[RETURN]
```

You should hear the disk drive give a short and quiet 'whirr' as the motorised hub starts up. A message line containing the words 'Lynx DOS', the version number and the current value of HIMEM will be displayed. The new '>' prompt indicates that the system is ready to accept DOS commands.



Open the disk drive door by pushing the door catch in. The door should spring upwards, leaving a open gap into which a floppy disk can be slotted.

Take a new disk, without any data on it and insert it into the drive. The disk should be handled as in shown in the diagram on the right, with the 'read slot' nearest the drive door.

Close the door gently but positively. The door catch will spring into a locked position.

SOME BASIC DOS COMMANDS

The DOS commands (and there are nineteen of them which are more fully described in the next Chapter), are extensions to BASIC - so they are entered by typing 'EXT ' before the command name. In this introductory session, you only need to use the following commands:

```
     EXT FORMAT
     EXT DIR
     EXT SAVE
     EXT LOAD
     EXT ERA
```

Let's try the first of these, to 'format' the empty disk which is in the disk drive. Type:

```
EXT FORMAT "A:"[RETURN]
```

The process usually takes less than one minute. Lynx disk drives are unusually silent, but you should hear a series of quiet clicks as the drive head skips from track to track.

If the disk has already been formatted for the Lynx, there will be the warning message:

```
Disk contains data
OK to continue? (Y/N)
```

because formatting completely "wipes" all existing data from the disk. If you do get this warning, then you will have to make up your own mind about the value of the data on the disk before giving 'Y[RETURN]' - (yes) as a go-ahead.

After you have a disk in the drive, formatted for use with the Lynx, the next task is to use the disk to 'save' a BASIC program. This use of a storage medium will be familiar to those who have used a cassette for the same purpose. Let's put a simple program in memory. You can write one of your own, or enter:

```
100 CLS
110 FOR J=1 TO 10
120   PRINT "Greetings from Diskland"
130 NEXT J
```

Now type:

```
EXT SAVE "TEST"[RETURN]
```

and wait for the BASIC prompt '>'. You will already appreciate the speed with which files can be saved on disk, compared to cassette tape!

Let's check that the file is on the disk by examining the disk 'directory', or list of file contents. Type:

```
EXT DIR
```

If you called your test program 'TEST', then the reply will be of the form:

```
6 blocks used, 794 blocks free
.
TEST
```

The first file listed is the directory itself. It is given the name '.' by the system. Now let's load and run that we have stored on disk, but to make sure it's coming from disk and not the Lynx's RAM, type:

```
     NEW[RETURN]
```

Then load the program from disk:

```
     EXT LOAD "TEST"[RETURN]
```

and run it:

```
     RUN[RETURN]
```

You should get the expected results. Store a copy of this program back to disk again, under a different name:

```
     EXT SAVE"TEST..2"[RETURN]
```

Check the directory again, if you wish.

Another useful command is provided by 'EXT ERASE', which allows you to delete files. Try deleting the first file that you saved, called 'TEST':

```
     EXT ERASE "TEST"[RETURN]
```

Now examine the directory:

```
     EXT DIR[RETURN]
```

Only the directory and the second file should be shown as being on the disk:

```
     6 blocks used, 794 blocks free
     .
     TEST..2
```

This brief session has shown you how to attach the hardware units together, how to power up the system correctly, and introduced you to some of the most used DOS commands: EXT FORMAT, EXT SAVE, EXT DIR, EXT LOAD and EXT ERASE.

POWERING DOWN

Before you end this session, however, you must know how to switch the system off safely.

A simple rule for protecting disks is that THE SYSTEM SHOULD ALWAYS BE POWERED UP BEFORE DISKS ARE INTRODUCED INTO THE DRIVES - AND DISKS SHOULD BE REMOVED BEFORE THE SYSTEM IS POWERED DOWN.

After the disks are removed (which, of course, should never be done before the system has completed its tasks with the disk), the Lynx is switched off, followed by the disk drive unit.

You are now ready to explore the rest of the DOS commands, in the next Chapter.

Chapter 4: THE DOS COMMANDS

The Lynx DOS adds a further 19 commands to those already available from BASIC. These are listed below and described more fully overleaf.

In many cases, what the commands actually do will be clear from their name alone, but others have more subtle uses. Some have been already been introduced in Chapter 3 of this manual ('Getting Started'). In addition, the operation of commands such as EXT SAVE and EXT LOAD will be familiar to those who have used a cassette recorder to save and load programs with the Lynx.

All Lynx DOS commands are extensions to BASIC and are preceded by '**EXT** '. The commands are listed below in alphabetical order, but without 'EXT ' in front of them, so as to make them more readable.

Commonly used commands like 'EXT SAVE' and 'EXT LOAD' have counterparts in the Cassette Operating System. There is a danger here in that if you forget the 'EXT ' before issuing the command, the Lynx will look to the cassette unit and not to the disk unit to carry out the instruction. If no cassette is connected, the machine will hang up. The only remedy for this is to restart from power up ('re-boot') which unfortunately loses any material you may have in RAM - so beware!

Remember that you can generate 'EXT ' by holding down the <ESC> key and typing in 'E'. This saves three keystrokes and you will find it less tedious in the long term.

Command	**Minimum abbreviation**
APPEND	**AP.**
BOOT	**BO.**
CHECK	**CHE.**
COMMENT	**CO.**
DCSAVE	**DC.**
DIR	**DI.**
DRIVE	**DR.**
DSLOAD	**DSL.**
DSSAVE	**DSS.**
ERASE	**ER.**
FORMAT	**FO.**
INFO	**IN.**
LOAD	**LO.**
LOCK	**LOC.**
MLOAD	**ML.**
MSAVE	**MS.**
RENAME	**RE.**
SAVE	**SA.**
UNERASE	**UN.**

EXT APPEND

This command allows you to add material stored in a file on disk to the end of a program stored in internal memory (RAM).

Say you have a useful subroutine in BASIC which you have stored on disk A: in a file called, for example, SUBROUTINE. Typing:

```
     EXT APPEND "A:SUBROUTINE"[RETURN]
```

would take the program on the disk in drive "A" and append it (that is, attach it to the end) of the BASIC program currently held in memory.

The first line number of the material that you are adding from disk must be higher than the last line number of the program already in memory. If this is not the case, the sequence of BASIC line numbers will appear reversed, or there may be duplicate line numbers; these problems may be remedied by using the BASIC 'RENUM' command.

In general, make sure that the line numbers of any file you might wish subsequently to append from disk ("A:SUBROUTINE" in the example above) is given high line numbers, much higher than those of any BASIC program you are likely to be working on in RAM. You can do this before storing the program on disk (which you would do with the EXT SAVE command) by using the 'RENUM' command.

EXT BOOT

The BOOT command reads and executes the "bootstrap" code from the specified drive. It is mainly designed to initialise the CP/M operating system, when this is available for the Lynx.

```
     EXT BO. "A:"[RETURN]
```

If no executable code is found on the disk in the selected drive, the error code B5 is displayed.

EXT CHECK

This command checks and verifies the layout of a disk. It also indicates the space already allocated and the amount of space still free on the disk. One use of 'check' is to discover if a certain disk has been formatted before using it.

```
     EXT CHECK "A:"[RETURN]
```

will return a response such as:

```
     DOS error A6:
```

in the case of a corrupt disk, an unformatted disk or even one formatted for another make of computer (see the 'EXT FORMAT' command). The more normal response, however, is:

```
     Directory OK
     5 blocks used, 795 blocks free
```

in the case of a freshly formatted disk.

EXT COMMENT

This command lets you add up to 32 characters of comment, annotation, explanation or date stamp and to permanently attach it to files saved on disk.

EXT COMMENT is used to establish one of these comment strings. Once set up in memory, it will be added to every file you subsequently save until power is withdrawn, a re-initialisation is carried out or a new comment string is set up. Comments are BASIC strings, so they must be enclosed in a pair of double quotation marks; characters after the 32nd are ignored.

Any comment saved with a file is displayed whenever a file inquiry is made (see the EXT INFO command). For example:

```
     EXT COMMENT "Friday 13th January"[RETURN]    or

     EXT CO. "Fractals program - Version 3.2"[RETURN]
```

will label all the files saved to disk after the comment string was set up with the given comment string. If nc comment is required, then you can replace the old comment with a new one containing a null entry, thus:

```
EXT COMMENT[RETURN]  or

EXT CO.[RETURN]
```

EXT DCSAVE

EXT DCSAVE is a variant of the EXT DSSAVE command. It saves the contents of a chain of stores to a named file on disk. For example:

```
EXT DCSAVE 5,"SOME.DATA"[RETURN]
```

will take the contents of RAM store number 5 and preserve it in the named file on the default disk drive.

The contents of several stores can be combined using the EXT CHAIN command (EXT CHAIN is not a DOS command, it also works with the Cassette Operating System). In the following example, the contents of stores 5, 6 and 7 are combined, and stored on disk as one continuous file:

```
EXT CHAIN 5,6[RETURN]
EXT CHAIN 6,7[RETURN]
EXT CHAIN 7,0[RETURN]    ('0' marks the end of a chain)

EXT DCSAVE 5,"LOTZA.DATA"[RETURN]
```

Associated commands: EXT DSSAVE and EXT DSLOAD.

EXT DIR

This is a way of displaying the disk usage and the list of filenames contained on a particular disk.

The first file listed will be ".", representing the file in which the directory itself is stored. To avoid corruption of the directory, this file is protected against deletion or access in the normal way. Thus:

```
EXT DIR "A:"[RETURN]     or

EXT D.[RETURN]
```

will first give the number of blocks free and used, followed by a list of the files currently stored on the disk in drive A. One file is listed per line, with "." appearing first. The second example will give the same result, for the default drive.

The display of the directory may be controlled by pressing the <SHIFT> key to pause output to the screen. The display resumes when the key is released. Pressing <ESC> abandons the directory listing.

EXT DRIVE

The EXT DRIVE command can be used if you have a Lynx with more than one disk drive. (Up to four are possible and these will be called drives A, B, C and D.) Unless otherwise specified, the Lynx assumes that the disk in drive A will be used. This is called the default drive. The EXT DRIVE command is used if you need to change the default drive at any time. For example:

```
     EXT DRIVE "B:"[RETURN]
```

uses the disk in drive B in a multiple drive system, until you change it (with EXT DRIVE) or reboot.

On a single disk drive system, this command will have no effect as the drive always defaults to drive A.

EXT DSLOAD

EXT DSLOAD adds the contents of the given disk file to the specified low RAM data store, numbered from 1 to 48. The file contents are added to the end of the store contents without changing or overwriting those store contents.

```
     EXT DSL. 1,"ADDSOME.DATA"[RETURN]
```

As with the corresponding tape commands EXT SSAVE and EXT SLOAD, the number of the data store is not saved along with the file. This means that store contents saved as a disk file can be loaded back into a data store with a different number. For example, you can:

```
     EXT DSSAVE 5,"SOME.DATA"[RETURN]
```

and later:

```
     EXT DSLOAD 42,"SOME.DATA"[RETURN]
```

Associated commands: EXT DSSAVE and EXT DCSAVE.

EXT DSSAVE

This command saves the contents of the given store to the named file on disk. ('Stores' are low RAM areas of variable size, numbered from 1 to 48.) For example:

```
     EXT DSSAVE 1,"B:FIRST.DATASTORE"[RETURN]
```

Associated commands: EXT DSLOAD and EXT DCSAVE.

EXT FORMAT

This command formats the specified disk. Formatting is the laying down of the tracks and sectors on the disk and it has to be carried out before a new disk can be used to store files.

The EXT FORMAT command can also be used to clear all the files from a disk if they are no longer needed. But make absolutely sure that this is the case before re-formatting (by using the EXT DIR command, for example), since there is no way of recovering the files lost in this way! To completely clear the disk in drive A for re-use, type in the command:

```
     EXT FORMAT "A:"[RETURN]  or

     EXT FO.[RETURN]
```

A completely blank (unformatted) disk, fresh from the manufacturer's box will format straight away; otherwise the response will be:

```
     Disk contains data
     OK to continue? (Y/N)
```

Enter the appropriate letter, followed by [RETURN]. If you type 'Y' (yes) the disk will be absolutely erased and an 'empty' disk with a fresh directory will be created.

EXT INFO

This command displays useful information about an existing file on a disk. For example:

```
     EXT INFO "A:FILE"[RETURN]
```

produces a display on the screen of the form:

```
Type=B          Lock=1          Blocks=4
Addr=&9000      Exec=&9035      Bytes=&0316
CJD test program 29/4/85
```

The components of the file information display are:

Type of file. Possible values are:
- B = BASIC program file
- M = Machine code
- D = Stored data (low 24K of RAM)

Lock gives the 'protection level' of the file. Values can range from 0 to 7; see the EXT LOCK command.

Blocks shows the number of blocks occupied by the disk file. Each complete block contains 256 bytes.

Addr, for files of type 'M' (machine code), shows the hexadecimal address in memory into which the file will be loaded by the EXT MLOAD command. This address is ignored by both EXT LOAD and EXT DSLOAD (which are used with files of type 'B' and 'D' respectively).

Exec gives the execution address in hexadecimal. This is only significant for 'B' and 'M' (BASIC and Machine code) files. A non-zero value means that the program will execute automatically (i.e. autorun) when loaded from disk. For machine code files the address given is the actual address at which execution will start. For BASIC files, Exec displays the offset by which Lynx DOS locates the starting line number; Exec=&0000 means that the file is not 'autorun'.

Bytes represents the number of bytes in the file (excluding the Lynx DOS header attached to each file). The value is in hexadecimal.

The last line of the EXT INFO display is a Comment of up to 32 characters. A comment string can be optionally set up before the file is saved. See the EXT COMMENT command for details.

EXT ERASE

This command removes the specified file from the disk directory. If the disk has not been written to in the meantime, it may be possible to restore an erased file by the EXT UNERASE command.

```
     EXT ERASE "A:OLD.PROGRAM"[RETURN]
```

Once deleted in this way, the file will no longer appear in the disk directory.

EXT LOAD

This command loads a BASIC program from disk into RAM. Any BASIC program in RAM at the time will be lost. For example:

```
     EXT LOAD "A:TEST.PROG"[RETURN]
```

would load the file TEST.PROG from disk A into internal memory.

In addition, a line number may be specified after the filename. For example:

```
     EXT LOAD "A:TEST.PROG",20[RETURN]
```

In this example, the file will load and automatically run the BASIC program called 'TEST.PROG' from line 20. A line number specified at the time of loading will override any line number given when the file was saved.

If you give a line number which is not present in the BASIC program, the loading into RAM will not take place; DOS error F5, with the message, 'Line not found' will be displayed.

In either case, don't forget the 'EXT ' before 'LOAD' - otherwise you will access the Cassette Operating System version of 'LOAD' and the Lynx will hang up (unless a cassette recorder is connected).

EXT LOCK

A 'lock' is a protection value given to a file. This means that you can prevent a file being erased, written to, or even read. A lock value is in the range 0 to 7 and is produced by adding together the levels of protection required. The levels are:

1 = locked against erasing
2 = locked against reading
3 = locked against writing

For example:

```
EXT LOCK "PRIVATE",6
```

would protect the file "PRIVATE" against being erased, read or written to (i.e. protection levels 1 + 2 + 3).

A newly saved file is given a lock value of 0, so no protection is in force. To 'unlock' a file, set the lock value back to zero:

```
EXT LOCK "PRIVATE",0
```

EXT MLOAD

If you want to load a machine code program or a block of binary data (that is, files of type 'M' as displayed by the EXT INFO command, described above) then use EXT MLOAD instead of EXT LOAD.

A filename must be given with this command.

Two further memory addresses may be optionally given as arguments. The first is a start address in memory to which the data is to be loaded from disk: the second is an execution address for the loaded block. The execution address overrides any such address specified when the file was saved (by the command EXT MSAVE, see below). For example:

```
EXT MLOAD "MCPROG"[RETURN]                or

EXT ML. "B:TEST",&9000[RETURN]
```

Associated commands: MSAVE

EXT MSAVE

This command saves a block of machine code or binary data (i.e. files of type 'M') to disk with the given filename. The start and end address of the block must also be supplied.

A third argument, the execution address, is optional. This gives the place in the code from which it will autorun when subsequently loaded back from disk. If a null address is given (that is, either no address at all or an address of '0'), then the code will not autorun when subsequently loaded back from disk. For example:

```
      MSAVE "A:MCPROG",&2000,&2100[RETURN]          or

      MS. "A:FRED",&1000,&1234[RETURN]
```

If you try to 'EXT MSAVE' a BASIC program (a type 'B' file) or a Stored data (a type 'D') file, then a "Wrong file type" message will be given.

Associated commands: MLOAD

EXT RENAME

This command is used when you wish to give a new name to a file already stored on disk.

```
      EXT RENAME "OLD.PROG","NEW.PROG"[RETURN]
```

Once a file has been renamed, the old file name will be replaced by the new name in the file directory. The file itself is unchanged.

Both the old file and the new file have to be on the same disk.

EXT SAVE

Saves a BASIC program currently in RAM as a file on disk. You can save a program so that it will automatically run as soon as it is loaded from disk by adding a line number to the ending of the EXT SAVE command. Thus:

```
      EXT SAVE "A:PROGRAM"[RETURN]
```

saves the file called 'Program'

```
      EXT SAVE "A:AUTOSTART",100[RETURN]
```

saves the file so that it will automatically run from line 10 when it is subsequently loaded.

As with the EXT LOAD command, don't forget the 'EXT ' before 'SAVE' - otherwise you will access the Cassette Operating System version of 'SAVE' and the Lynx will hang up (unless a cassette recorder is connected).

Note that you cannot save to an existing filename - it will produce 'DOS error 94: File already exists'. This is to protect the original file against being accidentally overwritten. If you have a new version of a file, use a different name (perhaps with the version number incorporated) or rename the old file before saving the new.

EXT UNERASE

If you accidentally delete a file it can be reinstated by using the EXT UNERASE command. However, this operation will only be possible if no other new files have been written to the disk, and no changes have been made to the directory (by EXT RENAME, for example) since the erasure of the file to be restored.

```
    EXT UNERASE "A:OLD.PROG"[RETURN]
```

XROM

If you are using a disk drive with your Lynx, then XROM is usually the first command that should be executed after powering up.

XROM is not itself a DOS command, but it is essential to the operation of all of the DOS commands. It clears the data store, copies the Disk Operating System from ROM into RAM and switches the ROM out. A fuller description of the initialisation routine is given in the Technical Appendix (Chapter 7).

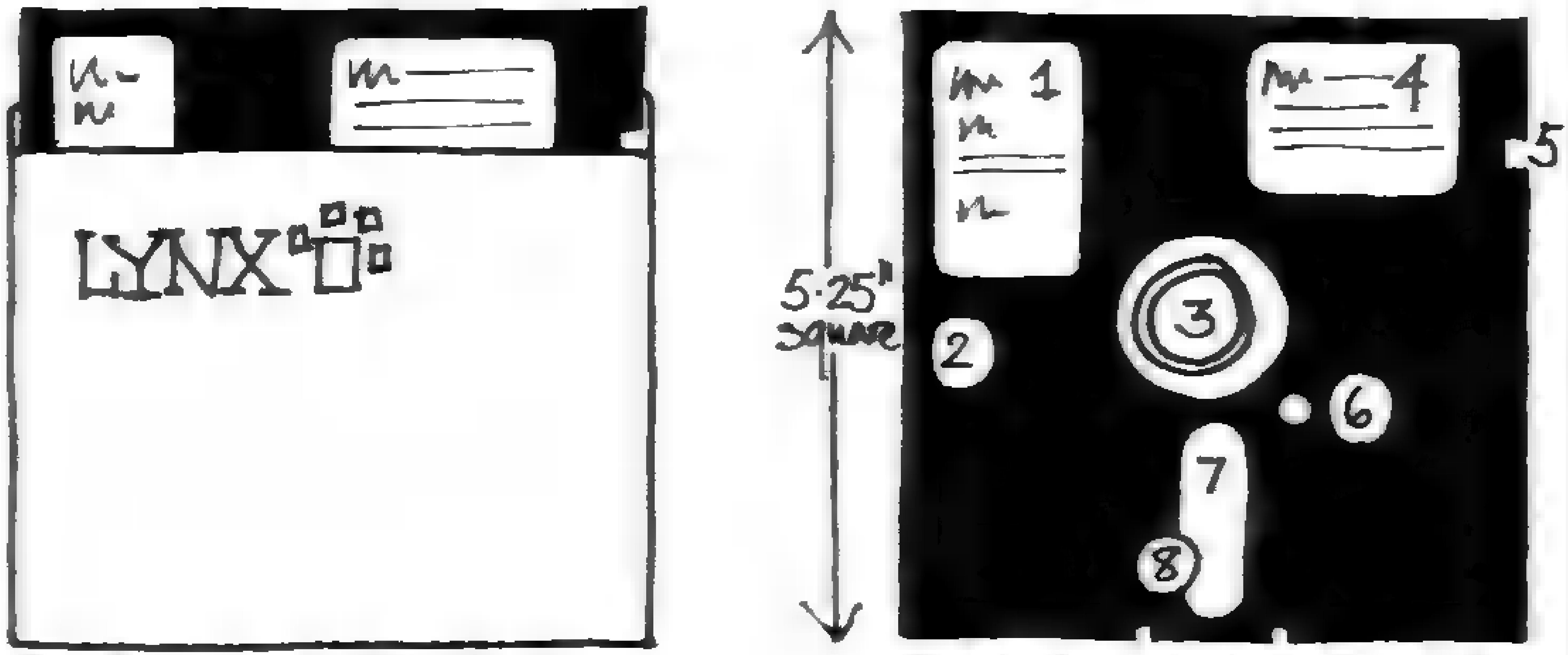
Note that 'XROM' is not an extension to BASIC, so you don't need to precede it with an 'EXT '.

```
    X.[RETURN]
```

Chapter 5: DISK CARE

LOOKING AFTER FLOPPY DISKS

Floppy disks are thin, round plastic sheets, the surface of which is coated with a magnetic compound physically similar to the coating on an audio cassette tape. The disk is permanently enclosed in a 5.25" square plastic jacket. The main features of a floppy disk (which is sometimes called a diskette or flexible disk) are shown in the diagram below:



The disk in its ENVELOPE

The disk in its protective jacket

1 Manufacturer's label
2 Protective plastic jacket
3 Drive spindle hole
4 Contents label (temporary)
5 WRITE PROTECT notch
6 Index hole
7 Disk Surface—DON'T TOUCH!
8 READ/WRITE head slot

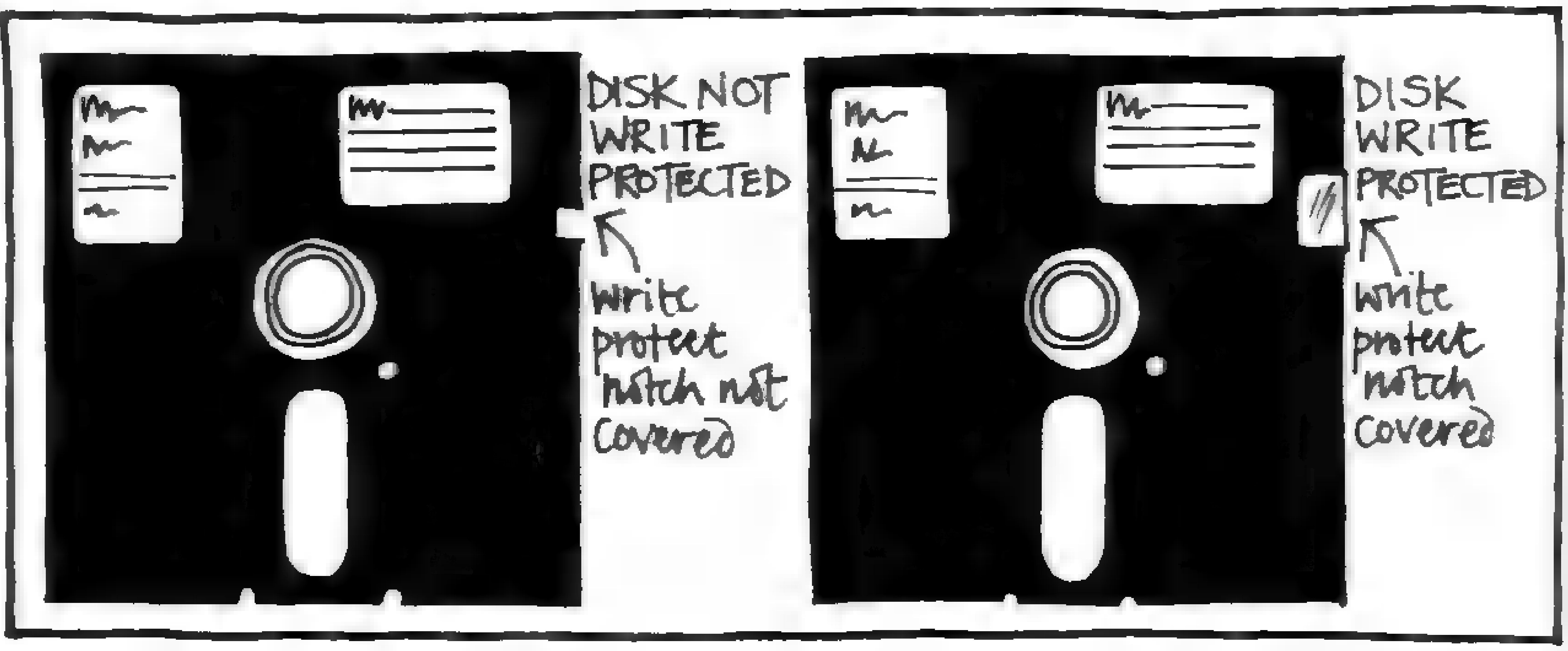
Disks should never be bent, folded or scratched. The exposed magnetic surface is very vulnerable to damage by greasy fingerprints (all fingerprints are greasy!) so don't touch the disk surface at all. The tiniest particles of food, drink or cigarette ash which come into direct contact with the disk surface can make it unusable, corrupting all the data the disk contained. Since each disk can contain hours, if not weeks of work, it is best not to smoke, eat or drink while handling them. To prevent this type of damage, every disk should be returned to its protective envelope as soon as it leaves the disk drive. Plastic library boxes which can organise and store many disks, and keep them dust-free, are recommended.

Never leave disks on or near a direct source of heat such as a radiator or storage heater, nor near a source of strong magnetic fields. The tops of television sets and monitors can be a source of high energy magnetic fields and a large temperature gradient - so they are bad places to leave your disks on both counts.

Floppy disks are supplied with 'low tack' removable adhesive labels for recording their contents. Always write the labels out before sticking them on to the disk - the impression left by ball-point pens can damage the disk surface. Soft felt-tip pens, if used carefully, may be used for writing on labels already applied.

It is reccommended that you always make copies of your valuable program or data files on a separate disk. This is in case the original disk is accidentally corrupted or mechanically damaged. A single 'archive disk' can often be used to back-up several partly filled 'working disks'.

WRITE PROTECTION



Some disks will contain important information, programs or data which you have no intention of writing over or deleting. In order to 'write-protect' all the files on a particular disk, all you need do is to cover over the 'write-protect notch' (on the top right-hand side of the disk) with a small adhesive label. Foil write protect labels are supplied with each new box of the disks.

An attempt to write to a 'write protected' disk (by operations such as 'EXT SAVE', for example), produces the DOS Error: C5.

RECOMMENDED DISKS

- For the 200K disk drive, the disk type required is 5.25" single sided, soft sectored, 40-track, double density (48 Tracks Per Inch).

- For the 800K, the specification is the same, except that it requires quad density (96 TPI), 80-track disks.

Ask your Lynx dealer to recommend suitable disks. It is generally worth buying high quality diskettes rather than a cheaper alternative. These are manufactured to much higher standards and are less likely to leave loose oxide particles on the read/write heads, or to suffer from 'dropouts' in recording quality. Some disks have reinforced hubs - the inner rim where the disk drive itself engages - and these will generally prove more reliable because they allow more regular alignment with the disk drive spindle.

Disks are sold in boxes of ten, (complete with temporary contents labels and 'write protect' foil labels,) but nearly all dealers will sell them individually if requested.

Disk drive head cleaning head kits are available. These normally comprise of lint-free non-abrasive diskettes and a cleaning solution. Cleaning should not need to be done very often (it depends on the environment and usage of the disk system) once a year should be adequate. It is better to prevent contaminants from accumulating near a disk head than to clean this precision mechanism too enthusiastically.

INITIALISATION ROUTINE

You are recommended to power the disk unit before switching on the Lynx (which must already have its disk drive expansion pack installed). Powering up switches in the disk ROM. Lynx DOS is initialised by the XROM command. This serves to:

Reinitialise the data store (existing data, if any, is lost)

Copy the DOS code from ROM into RAM. This code occupies 8 Kbytes - approximately 5.5 Kbytes of which is located in the data store area and 2.5 Kbytes in the top of RAM. This has the effect of reducing the data store area to about 15.5 Kbytes. The data store functions as before to the user, but there is less room for storage.

Switch out the ROM (containing the DOS)

HIMEM is moved up, to recover some of the RAM area previously masked out by the disk ROM.

FILENAMES

Programs are stored on disk as a variable length stream of bytes known as files. Files are referred to for reading and writing purposes by a 'drive name' (valid names are A:, B:, C: and D:). This is followed by a BASIC string of characters (letters, numbers or full-stops) which is the 'filename'.

A filename may have up to 28 characters, with upper and lower case letters being treated as identical. Filenames longer than this are treated as invalid by the DOS command interpreter. The filename is a BASIC string, so a BASIC variable may be freely mixed with parts of filename strings if this is advantageous. For example:

```
LET F$="Test.Module..."

EXT SAVE F$+"3"
```

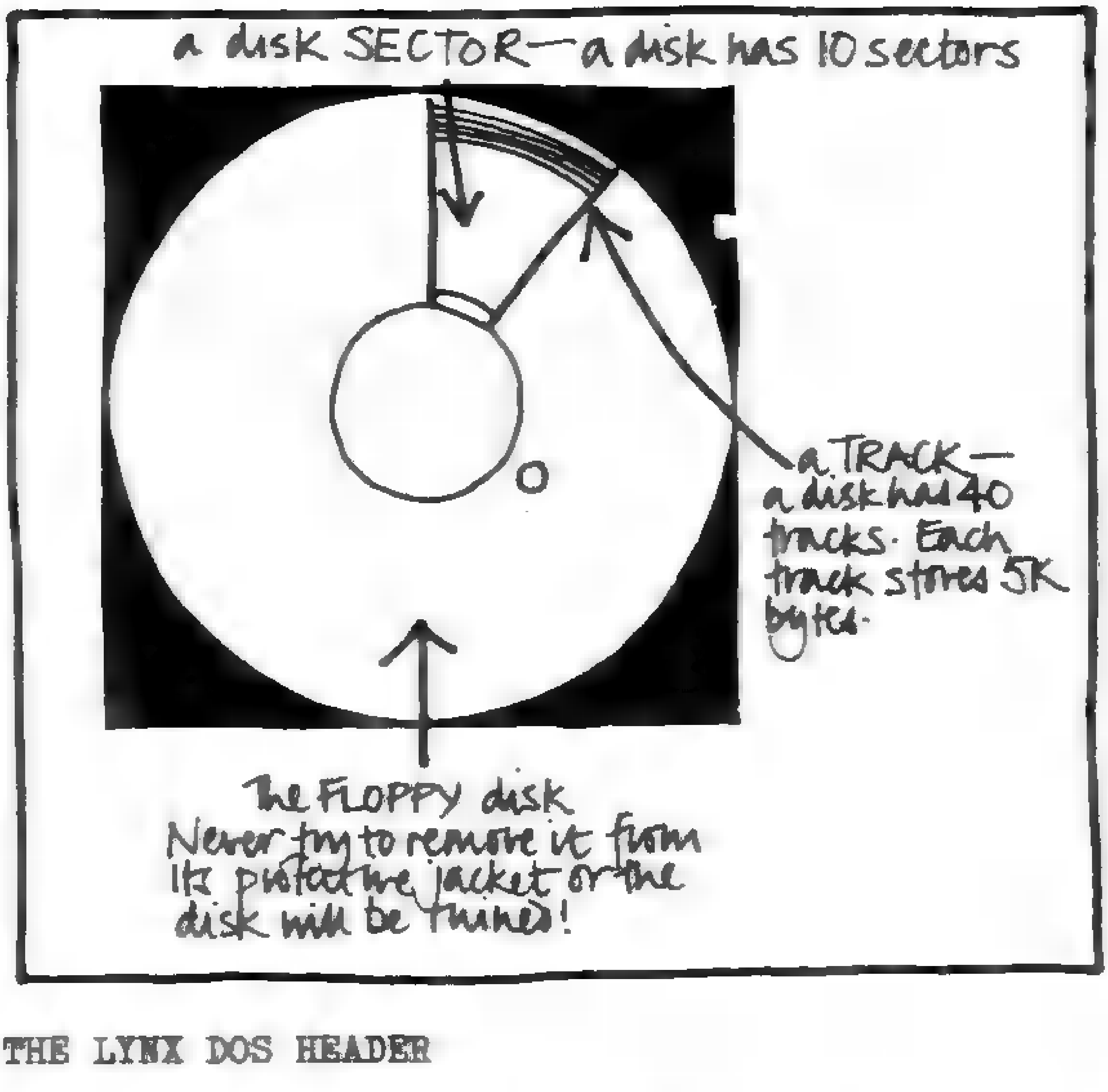
If the disk drive specifier is omitted, a default disk drive is assumed. The drive letter (and accompanying colon) is not part of the 'filename' and is not stored on disk. You need to distinguish a particular disk drive (by its letter and colon) in a multiple drive system.

THE DISK DIRECTORY

The directory of an individual disk is stored as special file with a special filename, which consists of a single full stop (for example, "A:."). The directory file can't be opened or read in the normal manner; you see its contents when you use the EXT DIR command. The directory file is protected from accidental deletion - although it is permanently removed by reformatting the disk, of course.

Individual directory entries are 32 bytes long. The first 28 bytes represent the filename, in normal ASCII code. The trailing four bytes are reserved for use by the DOS.

THE LYNX DISK STRUCTURE


a disk SECTOR— a disk has 10 sectors
a TRACK— a disk has 40 tracks. Each track stores 5K bytes.
O
The FLOPPY disk
Never try to remove it from its protective jacket or the disk will be ruined!

THE LYNX DOS HEADER

The header is a set of information given in the first 64 bytes of each file stored on disk. The information is stored in a consistent format and is used by commands such as EXT INFO, which provide a convenient presentation of the data.

The Lynx DOS header is not accessible via BASIC programs.

Offset	Contents
0	Must be 96H, otherwise a "Wrong file type" message will be issued.
1	An ASCII capital letter showing the file type. Lynx DOS creates files of types 'B', 'M' and 'D', but any letter can be read back.
2-3	Length of file excluding the header (low byte first, high byte second).
4-5	Start address of file, low byte first.
6-7	Execution address, again, with the low byte first. For a BASIC program, this is a negative offset from the end of BASIC to the line at which execution will start.
8-31	Not used by Lynx DOS
32-63	Comment supplied by the user. Any characters are allowed (see the EXT COMMENT command).

CP/M - AN ALTERNATIVE DOS

CP/M is an alternative to Lynx DOS which is recognised as the industry standard Disk Operating System for 8-bit microcomputers.

The main advantage of CP/M is that it gives the user access to a large body of existing software - licensed through Camsoft. The system also allows a limited transfer of programs between most Z80-based machines.

The main difference between Lynx DOS and CP/M (apart from differences in the names of commands and the range of facilities offered) is the fact that Lynx DOS is intimately connected with the Lynx BASIC interpreter. CP/M provides an operating environment for many other specialised languages besides BASIC. Programming languages (including fast compiled languages with advanced features and specialist capabilites), sophisticated applications programs (such as word processing, 'spreadsheet' or accounting packages) have all been implemented under CP/M.

CP/M on the Lynx will provide a gateway to enormous amounts of tested and professionally developed 'serious' and business software.

Chapter 7: ERROR MESSAGES AND ERROR CODES

SYNTAX ERRORS

For the newcomer to Lynx DOS, the most common sign that a disk command has not been successful will probably be the message:

> Syntax Error

This simply means that the Lynx could not interpret your request and you will have to retype it. 'Syntax errors' can often be the result of mistyping, of omitting required parameters (leaving off a drive specifier, say) or forgetting a quotation mark around a filename string - in fact anything that doesn't meet the simple grammar of DOS commands will give a syntax error. Check the command descriptions and try out the exact examples in Chapter 4.

If an error occurs, Lynx DOS will display message 'DOS error ' followed by a two digit (hexadecimal) error number and colon. For the errors most likely to occur in practice, the error code is accompanied by a brief error message.

ERROR MESSAGES

File not found

The file from which you are trying to read is not on the disk. This is often the result of mistyping the filename. Do a directory listing (using the EXT DIR command).

Wrong file type

This could be an attempt to EXT LOAD a machine code program (i.e. using an 'M' type file for a 'B' type file operation). It could also arise from not using a Lynx DOS file at all.

Out of memory

Perhaps the file is too large to be loaded into free RAM or there is not enough space for the format buffer.

File locked

The lock status of the file will not allow the attempted operation. Use EXT LOCK to change the lock status.

Drive not ready

The disk drive door is open or there is no disk in the drive.

Disk full

The standard disk allows 800 blocks (five of which are used for the minimum directory). This equals 200 Kbytes of formatted data.

ERROR CODES

The hexadecimal Error Codes themselves are given here for completion:

80	Invalid channel number
81	Invalid drive specifier
82	Invalid file name
83	Invalid function code
84	Invalid sub-function code (to function 11)
85	Invalid parameter
86	File or disk unexpectedly open
87	No file or disk file left open
90	Not a Lynx disk
91	Invalid disk type for this system
92	No directory created on disk
93	File not found in directory
94	File already exists
A0	Disk full
A1	Directory full
A2	Too many files
A3	File reached maximum size (64 Kbytes less header)
A4	Protection error
A5	System error (disk corrupted)
A6	Internal error (should not occur)
B0	LOAD - end of file
B1	RENAME - attempt to access more than one disk drive
B2	UNERASE - file was not recoverable
B3	FORMAT - Invalid format type
B4	FORMAT - Verification error
B5	BOOT - No executable code on disk
C0	Disk error - No disk in drive
C1	Disk error - Lost data
C2	Disk error - CRC (check) error
C3	Disk error - Record not found or seek error
C4	Disk error - Write fault
C5	Disk error - Disk is write protected
C6	Disk error - Drive not ready

LYNX DOS REFERENCE CARD

Command	Minimum abbreviation <ESC>+E +
EXT APPEND <filename> [,<line_number>]	**A.**
EXT BOOT [<drive>]	**BO.**
EXT CHECK [<drive>]	**CHE.**
EXT COMMENT [<comment>]	**CO.**
EXT DCSAVE <store>,<filename>	**DC.**
EXT DIR [<drive>]	**D.**
EXT DRIVE <drive>	**DR.**
EXT DSLOAD <store>,<filename>	**DSL.**
EXT DSSAVE <store>,<filename>	**DSS.**
EXT ERASE <filename>	**ER.**
EXT FORMAT [<drive>]	**FO.**
EXT INFO <filename>	**IN.**
EXT LOAD <filename> [,<line_number>]	**LO.**
EXT LOCK <filename>,<value>	**LOC.**
EXT MLOAD <filename> [,<start_addr>] [,<exec_addr>]	**ML.**
EXT MSAVE <filename>,<start_addr>,<end_addr> [,<exec_addr>]	**MS.**
EXT RENAME <old_filename>,<new_filename>	**R.**
EXT SAVE <filename> [,<line_number>]	**SA.**
EXT UNERASE <filename>	**U.**
XROM	**X.**

<filename> = a BASIC string of up to 28 characters, comprised of letters, numbers or "."
<line_number> = a line number used in a BASIC program
<drive> = a disk drive name, either "A:", "B:", "C:" or "D:"
<store> = numbered low RAM store, from 1 to 48
<start_addr>
<end_addr>
<exec_addr> = Hexadecimal addresses
<lock_value> = a protection status number, in the range 0 to 7
<comment> = a BASIC string of up to 32 characters

Items enclosed in square brackets are optional and can be omitted.